Constrained Optimization in Delocalized Internal Coordinates

JON BAKER

Department of Chemistry and Biochemistry, University of Arkansas, Fayetteville, Arkansas 72701

Received 9 June 1996; accepted 19 November 1996

ABSTRACT: Using the recently introduced delocalized internal coordinates, in conjunction with the classical method of Lagrange multipliers, an algorithm for constrained optimization is presented in which the desired constraints do *not* have to be satisfied in the starting geometry. The method used is related to a previous algorithm by the same author for constrained optimization in Cartesian coordinates [*J. Comput. Chem.*, **13**, 240 (1992)], but is simpler and far more efficient. Any internal (distance or angle/torsion) constraint can be imposed between any atoms in the system whether or not the atoms involved are formally bonded. Imposed constraints can be satisfied exactly. *J Comput Chem* **18**: 1079–1095, 1997

Keywords: geometry optimization; constraints; delocalized internal coordinates

Introduction

It is now firmly established that natural internal coordinates[1, 2] are perhaps the best choice for efficient geometry optimization in molecular systems.[3] For minimization, natural internals can be used effectively even without preliminary Hessian data [3]. Cartesian coordinates, because of their much greater coupling, are very inefficient without reliable second derivative information.[3, 4] The efficiency of natural internal coordinates with respect to Cartesians—in terms of the number of cycles required to reach convergence—increases with system size.

Correspondence to: J. Baker

Recently a ''new'' set of coordinates—delocalized internal coordinates—has been introduced.[5] They are related to natural internal coordinates, being derived from a set of individual (primitive) stretches, bends, and torsions, which are themselves generated by a ''blanket approach'' based on the atomic connectivity. Delocalized internals form a complete, nonredundant coordinate set and can be generated automatically for essentially *any* molecule, regardless of topology. No user input is required to define them. Their construction and use is straightforward and is based on ideas first introduced in this context in the redundant optimization scheme of Pulay and Fogarasi.[6]

The article introducing delocalized internal coordinates [5] also showed how, using an appropriate Schmidt-orthogonalization procedure, indi-

 CCC 0192-8651 / 97 / 081079-17

vidual primitives can be isolated within the coordinate set and constraints subsequently imposed by eliminating the isolated coordinates. Any internal constraint—that is, any interatomic distance, angle, torsion, or out-of-plane bend—can be imposed between any atoms in the molecule, regardless of whether the atoms involved are formally bonded. Imposed constraints can be satisfied exactly. Again, no user input—other than information as to which constraints to impose—is required. The method works very well and is extremely powerful.[5]

There is, however, one weakness in the constrained optimization scheme using delocalized internal coordinates introduced in ref. 5: The desired constraint *must* be satisfied in the starting geometry. Actually, this is not really a weakness at all compared to currently popular methods for constrained optimization in quantum chemistry—the same is true, for example, for standard *Z*-matrix optimization. However, with the increasing use of graphical molecular model builders, which ultimately produce the molecular geometry as a set of Cartesian coordinates, and increasing computational power leading to calculations on considerably larger systems than were viable even a few years ago, the *Z*-matrix is much less useful today than it has been in the past.

A fairly efficient algorithm for constrained optimization capable of treating constraints that are *not* satisfied in the initial geometry has in fact already been introduced by this author.[7] This was developed precisely for the situation referred to above, namely imposing constraints on a molecule obtained from a graphical model builder as a set of Cartesian coordinates. The algorithm was written in early 1991, before the widespread use and acceptance of natural internal coordinates, and carries out constrained optimization in Cartesian coordinates, using either penalty functions[8] or Lagrange multipliers.[8,9]

Although this algorithm is reasonably efficient in the context of Cartesian optimization, it would clearly be highly desirable if initially unsatisfied constraints could be handled directly in internal coordinates, because these are far superior to Cartesians for general optimization. This is exactly what is done in this study. Using a modification of the Lagrange multiplier algorithm presented in ref. 7, and incorporating this into the delocalized internal coordinate scheme introduced in ref. 5, an algorithm for constrained optimization directly in internal coordinates is presented, which can handle, simply and efficiently, constraints that are not satisfied in the starting geometry.

The article is organized as follows. A brief review of the theoretical background to both delocalized internal coordinates and the Lagrange multiplier algorithm is given in the next section. The third section discusses the implementation of the Lagrange multiplier algorithm in delocalized internals using fluoroethylene as an illustrative example. The fourth section presents the results of several constrained optimizations, contrasting the performance of the original Cartesian algorithm with the constrained optimization algorithm in delocalized internals. The final section comprises a summary and conclusions.

Theoretical Background

DELOCALIZED INTERNAL COORDINATES

Delocalized internal coordinates are generated from an initial set of primitive valence coordinates used to describe the geometry of the molecule. There should be a sufficient quantity of these primitives to span all the degrees of freedom of the system under consideration. The method used to obtain the initial set of primitives in ref. 5 is to take all stretches (bond lengths), all planar bends, and all proper torsions based on the atomic connectivity. This ''blanket approach'' usually generates far more primitives than are necessary to describe the (typically) $3N - 6$ degrees of freedom of a molecule containing N atoms.

Displacements in internal coordinates are related to the equivalent displacements in Cartesians (for small displacements) by the well-known B matrix.[10] Although the set of primitive internals will have many redundancies, the full B matrix is formed for all n, say, primitive coordinates. We then form and diagonalize the $n \times n$ matrix $\mathbf{G} = \mathbf{B}\mathbf{B}^{\mathbf{T}}$. Diagonalization of $\mathbf{G}$ results in two sets of eigenvectors; a set of $m = 3N - 6$ eigenvectors with eigenvalues $\lambda > 0$ (the *nonredundant* or *active* set, $\mathbf{U}$) and a set of $n - m$ eigenvectors with eigenvalues $\lambda = 0$ (the *redundant* set, $\mathbf{R}$). The eigenvalue equation for $\mathbf{G}$ can be written:

$$\mathbf{G}(\mathbf{UR}) = (\mathbf{UR})\begin{pmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (1)$$

This natural separation into two vector subspaces allows us to reject the redundant part of our primitive space, retaining only the nonredundant set of vectors, $\mathbf{U}$. These eigenvectors, $\mathbf{U}$, from which

further vectors may be eliminated by symmetry considerations, constitute the set of *delocalized internal coordinates* used to carry out the subsequent geometry optimization.

Note that each vector in **U** (each delocalized internal coordinate) is a linear combination of potentially *all* of the original primitive internals. Using an appropriate Schmidt-orthogonalization procedure, it is possible to isolate any individual primitive into a *single* vector, and by eliminating this vector from the optimization space, effectively impose a constraint on that particular primitive. The given primitive will keep the same value throughout the optimization as it had in the starting structure, when it was isolated and eliminated. However, constraints *cannot* be imposed by elimination if the isolated coordinate does not have the desired value in the starting structure. For full details, see ref. 5.

Very recently, Schlegel and coworkers have introduced a redundant internal coordinate optimization scheme into GAUSSIAN 94,[11] and some comments are perhaps in order as to the relationship between the original redundant internal coordinate approach of Pulay and Fogarasi,[6] Schlegel's scheme,[11] and delocalized internal coordinates.[5] Note that ref. 5 already discusses some aspects of this, and further comments are made in an article by Baker and Pulay on optimization using redundant inverse-power distance coordinates.[12] However, because both referees of this article raised this issue, some (hopefully) clarifying remarks are given next.

First, it should be realized that, whatever internal coordinate scheme is used, gradients and possibly Hessians—which are calculated in Cartesians—have to be transformed into internal coordinates, and this involves taking the inverse of the **G** matrix. (Full details of the transformations required are given elsewhere.[5,6]) If the coordinate system contains redundancies, the **G** matrix is singular and thus cannot be directly inverted. Pulay and Fogarasi get around this problem[6] by using the concept of a *generalized inverse*, denoted by $\mathbf{G}^-$, where:

$$\mathbf{G}^- = (\mathbf{UR}) \begin{pmatrix} \Lambda^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{R}^T \end{pmatrix} \qquad (2)$$

that is, by transforming **G** to diagonal form, inverting only the *nonzero* eigenvalues, and transforming it back. Note that $\mathbf{G}^-$ is essentially $\mathbf{U}\Lambda^{-1}\mathbf{U}^T$; that is, only the nonredundant eigenvectors of **G** are used, but the transformation in eq. (2) effectively casts the optimization in terms of the *full* original coordinate set, including the redundancies.

A major disadvantage of keeping the full original (redundant) coordinate set is that the transformation in eq. (2) has to be done at the beginning of every optimization cycle. Even worse is that the backtransformation (the transformation of a new geometry in internal coordinates back to Cartesians), which is accomplished iteratively, requires essentially the same transformation at *each* iterative cycle. This may be very time-consuming, and can become prohibitively expensive as the system size increases for the more empirical Hamiltonians.[12]

In Pulay and Fogarasi's redundant internal coordinate scheme[6] the original set of coordinates were *natural internals*, the construction of which had already removed most of the redundancies among the primitive internals, and the redundant optimization scheme was used simply to remove any (small) remaining redundancy among the natural internals themselves. The size of the redundant optimization space in this approach is only slightly greater than that for a fully nonredundant optimization.

The key point recognized by both Schlegel and coworkers[11] and Baker et al.[5] was that the prior generation of natural internals (which requires thousands of lines of code to handle all the different topological possibilities) was unnecessary, and eq. (1) can be used directly with the individual primitive internals. However, Schlegel adopted the *same* scheme as Pulay and Fogarasi, simply replacing natural internals by primitive internals,[11] whereas Baker et al.—following the initial diagonalization of the **G** matrix—discarded the redundant coordinates entirely, retaining only the nonredundant coordinates, **U**.[5] The two approaches are related—indeed, the first cycle should give identical optimization steps—but differ significantly in the additional workload required, with Schlegel's scheme requiring more effort for a given initial primitive coordinate set.

The extra work involved in a straightforward adaptation of Pulay and Fogarasi's scheme to primitive internals arises because of the large increase in dimensionality, as the primitive set contains many redundancies. The inversions of the **G** matrix required on every optimization cycle soon become the most time-consuming part of the entire optimization algorithm. In Baker et al.'s approach, the *same* set of coordinates, **U**, are used throughout the entire optimization; there is no need to

solve eq. (1) at the beginning of each cycle (only *one* solution, at the start of the optimization, is required) and there is no need to construct the generalized inverse, given by eq. (2), as the coordinate set, **U**, contains *no* redundancies, and a normal inverse can be used for the transformations.

Perhaps the key to recognizing the difference between the three approaches is to ask which coordinate system is actually used to carry out the optimization. All three methods *start* from a set of primitive internal coordinates, typically containing many redundancies. In Pulay and Fogarasi's scheme,[6] these primitives are first used to construct a set of natural internal coordinates and these natural internals are then the *original* coordinates used in the redundant optimization scheme. Schlegel's scheme[11] is essentially identical to Pulay and Fogarasi's, except that it uses the primitive set *directly*, with the larger dimensionality that this entails. In Baker et al.'s approach, the primitives are used to construct a set of delocalized internal coordinates, **U**, and *these* then become the original coordinate set. Because the set **U** is fully nonredundant, technically, Baker et al.'s scheme is *not* a redundant optimization.

Ref. 11 also briefly discusses constrained optimization. This is handled using projection techniques which are generally much less efficient than Lagrange multipliers.[7]

LAGRANGE MULTIPLIER ALGORITHM

One of the best methods for imposing constraints that are *not* satisfied initially is the classic method of Lagrange multipliers. In the following review of this method, we follow ref. 7.

Without loss of generality we can consider the problem as that of minimizing a function of n variables, $F(\mathbf{x})$, subject to a series of m constraints of the form $C_i(\mathbf{x}) = 0, i = 1 \ldots m (m < n)$. We introduce the Lagrangian function:

$$L(\mathbf{x}, \lambda) = F(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i C_i(\mathbf{x}) \tag{3}$$

which replaces the function $F(\mathbf{x})$ in the unconstrained case. The λ_i are the Lagrange (unknown) multipliers, one for each constraint $C_i(\mathbf{x})$. Differentiating eq. (3) with respect to $\mathbf{X}$ and λ gives:

$$dL(\mathbf{x}, \lambda)/dx_j = dF(\mathbf{x})/dx_j - \sum_{i=1}^{m} \lambda_i \, dC_i(\mathbf{x})/dx_j \tag{4a}$$

$$dL(\mathbf{x}, \lambda)/d\lambda_i = -C_i(\mathbf{x}) \tag{4b}$$

At a stationary point of the Lagrangian function $\nabla \mathbf{L} = \mathbf{0}$, that is all $dL(\mathbf{x}, \lambda)/dx_j = 0$ and—in particular—all $dL(\mathbf{x}, \lambda)/d\lambda_i = 0$, which means that all constraints are satisfied. So, finding a set of values $(\mathbf{x}, \lambda)$ for which $\nabla \mathbf{L} = \mathbf{0}$ will give a solution to the constrained optimization problem in exactly the same way that a set of values $\mathbf{x}$ for which $\nabla \mathbf{F} = \mathbf{0}$ gives a solution to the unconstrained problem.

The Lagrangian second derivative matrix—the equivalent of the Hessian matrix in an unconstrained optimization—is given by:

$$\nabla^2 \mathbf{L} = \begin{pmatrix} d^2L(\mathbf{x}, \lambda)/dx_j \, dx_k & d^2L(\mathbf{x}, \lambda)/dx_j \, d\lambda_i \\ d^2L(\mathbf{x}, \lambda)/dx_j \, d\lambda_i & d^2L(\mathbf{x}, \lambda)/d\lambda_i \, d\lambda_j \end{pmatrix} \tag{5}$$

where:

$$d^2L(\mathbf{x}, \lambda)/dx_j \, dx_k = d^2F(\mathbf{x})/dx_j \, dx_k - \sum_{i=1}^{m} \lambda_i \, d^2C_i(\mathbf{x})/dx_j \, dx_k \tag{5a}$$

$$d^2L(\mathbf{x}, \lambda)/dx_j \, d\lambda_i = -dC_i(\mathbf{x})/dx_j \tag{5b}$$

$$d^2L(\mathbf{x}, \lambda)/d\lambda_i \, d\lambda_j = 0 \tag{5c}$$

Thus, in addition to the standard gradient vector and Hessian matrix for the *unconstrained* function $F(\mathbf{x})$, we need both the first and second derivatives (with respect to coordinate displacement) of the constraint functions. Once these quantities are available, the corresponding Lagrangian gradient —given by eq. (4)—and Lagrangian second derivative matrix—given by eq. (5)—can be formed, and the optimization step calculated in a similar manner to that for a standard unconstrained optimization.[7]

In the Lagrange multiplier method, the unknown multipliers, λ_i, are an integral part of the parameter set. This means that the optimization space consists of all n variables, $\mathbf{x}$, *plus* all m Lagrange multipliers, λ, one for each constraint. The total dimension of the constrained optimization problem, $n + m$, has thus increased by m compared to the corresponding unconstrained case. The Lagrangian Hessian matrix, $\nabla^2 \mathbf{L}$, has m extra modes compared to the standard (unconstrained) Hessian matrix, $\nabla^2 \mathbf{F}$. What normally happens is that these additional modes are dominated by the constraints (i.e., their largest components correspond to the constraint Lagrange multipliers) and they have *negative* curvature (a negative Hessian

eigenvalue). This is perhaps not surprising when one realizes that any motion in the parameter space that *breaks* the constraints is likely to *lower* the energy.

Compared to a standard unconstrained minimization, where a stationary point is sought at which the Hessian matrix has all positive eigenvalues, in the constrained problem we are looking for a stationary point of the Lagrangian function at which the Lagrangian Hessian matrix has as many negative eigenvalues as there are constraints; that is, we are looking for an *mth order saddle point*. For further details and practical applications of constrained optimization using Lagrange multipliers in Cartesian coordinates, see ref. 7.

Lagrange Multipliers in Delocalized Internal Coordinates

As previously mentioned, the original Lagrange multiplier algorithm for constrained optimization was written for Cartesian coordinates.[7] Its implementation in internal coordinates brings several simplifications and advantages.

In Cartesians, standard internal constraints—bond distances, angles, and torsions—are rather complicated nonlinear functions of the x, y, and z coordinates of the atoms involved. A torsion, for example, which involves four atoms, is a function of 12 different coordinates. In internals, on the other hand, each constraint is a coordinate in its own right and is therefore a simple linear function of just *one* coordinate (itself).

If we denote a general internal coordinate by R, then the constraint function, $C_i(\mathbf{R})$, is a function of one coordinate, R_i, and it and its derivatives can be written:

$$C_i(R_i) = R_i - R_0 \tag{6a}$$

$$dC_i(R_i)/dR_i = 1; \qquad dC_i(R_i)/dR_i = 0 \tag{6b}$$

$$d^2C_i(R_i)/dR_i\,dR_j = 0 \tag{6c}$$

where, in eq. (6a), R_0 is the desired value of the constrained coordinate, and R_i is its current value.

From eq. (6b), we see that the constraint normals, $dC_i(\mathbf{R})/dR_i$, are simply unit vectors and the Lagrangian Hessian matrix, eq. (5), can be obtained from the normal Hessian matrix by adding m columns (and m rows) of, again, unit vectors.

A further advantage, in addition to the considerable simplification, is the handling of 0° and 180° dihedral angle constraints. In Cartesian coordinates it is not possible to formally constrain bond angles and torsions to exactly 0° or 180° because the corresponding constraint normal is a zero vector. Such an attempt would result in zero Lagrangian Hessian eigenvalues because an entire row/column of the Hessian would be zero. From a practical point of view the situation for torsions is actually worse than this, because attempts to impose constraints near these limiting values often gives rise to oscillatory behavior which hinders convergence.[7] (This problem was solved later by introducing a dummy atom to break the constraint into a sum of two separate constraints; for instance, a 180° torsion could be reformulated as two separate 90° torsions, both of which were constrained.[13]) Similar difficulties do not arise in internal coordinates, at least for torsions, because the constraint normals are unit vectors regardless of the value of the constraint; thus, 0° and 180° dihedral angle constraints can be imposed just as easily as any other value. Bond angles of 180° still cause problems, but near-linear arrangements of atoms require special treatment even in unconstrained optimizations; a typical solution involves replaced a near 180° bond angle by two special linear coplanar and perpendicular bends,[14] and modifying the torsions where necessary. A linear arrangement can be enforced by constraining the coplanar and perpendicular bends.

One other advantage over Cartesians is that, in internals, the constraint coordinate can be eliminated once the constraint is satisfied to the desired accuracy (the default tolerance is 10^{-6} in atomic units—bohrs and radians). This cannot be done in Cartesians due to the functional form of the constraint. In Cartesians, therefore, the Lagrange multiplier algorithm must be used throughout the entire optimization, whereas in delocalized internal coordinates it need only be used until all desired constraints are satisfied; as constraints become satisfied they can simply be eliminated from the optimization space and once all constraint coordinates have been eliminated standard algorithms can be used in the space of the remaining unconstrained coordinates. Normally, unless the starting geometry is particularly poor in this regard, constraints are satisfied fairly early on in the optimization (and at more or less the same time for multiple constraints), and Lagrange multipliers only need to be used in the first few cycles of a constrained optimization in internal coordinates. On the first cycle of a constrained optimization all Lagrange multipliers are assumed to be zero.

To clarify the previous discussion we present an actual example, namely a constrained optimization on fluoroethylene. (The same molecule was also used as a pedagogical example in ref. 5.) Relevant printout from the current code for the first optimization cycle is shown in Scheme 1.

Three constraints are imposed on (planar) fluoroethylene: a C—C bond distance of 1.5 Å; a nonbonded H—H distance of 2.5 Å; and a ∠CCH angle of 123° (none of which are satisfied in the starting geometry). The initial set of primitive internals comprises five stretches, six bends, and

CONSTRAINED OPTIMIZATION IN DELOCALIZED INTERNAL COORDINATES
Searching for a Minimum

Optimization Cycle: 1

		Coordinates (Angstroms)		
ATOM		X	Y	Z
1	c	−.061684	.673790	.000000
2	c	−.061684	−.726210	.000000
3	f	1.174443	1.331050	.000000
4	h	−.927709	1.173790	.000000
5	h	−.927709	−1.226210	.000000
6	h	.804342	−1.226210	.000000

Point Group: cs Number of degrees of freedom: 9

Constraints and their Current Values

				Value	Constraint
Distance:	1	2		1.400000	1.500000
Distance:	4	5		2.400000	2.500000
Angle:	1	2	6	120.000	123.000

Attempting to Generate Delocalized Internal Coordinates
Generating Primitive Internal Coordinates
Primitive Stretches:

2	1
3	1
4	1
5	2
6	2

There are 5 Stretches
Primitive Bends:

2	1	3
2	1	4
3	1	4
5	2	1
6	2	1
5	2	6

There are 6 Bends
Primitive Torsions:

5	2	1	3
5	2	1	4
6	2	1	3
6	2	1	4

SCHEME 1.

There are 4 Torsions
There are 15 Primitive Internals
Adding Distance 4 5 to Primitive Set in order to Constrain it
There are now 16 Primitive Internals

Generating Delocalized Internal Coordinates

Eigenvalues of B*B(t):

.000000	.000000	.000000	.000000	.279460	.629472
.891576	.954679	1.155581	1.921891	2.022821	2.413512
2.784837	4.089852	4.335745	4.712469		

There are 4 Redundant primitive internals
Eliminated 3 Coordinates due to Symmetry

Final Set of Delocalized Internal Coordinates

−.482944	.137611	.137068	−.047913	−.431601	−.022264
−.137622	−.223061	.650914	−.086953	−.130908	.313941
.025380	.390655	.470961	−.082017	.317644	.359719
−.088953	−.328924	.014989	−.595144	.121322	−.328788
−.094837	.372854	−.081026	−.591590	−.224808	−.195127
−.582578	−.325494	−.127593	.003625	.160478	.177830
.213065	.275988	−.310086	.023676	−.119063	.269396
.369513	.049506	.437679	−.027301	−.041415	−.447225
.149168	−.376514	.080338	.222028	−.196324	−.152524
−.365674	.447350	.047111	.208152	.115494	−.278872
.216506	−.070836	−.127449	−.430180	.080830	.431396
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.078082	.033961	.004048	.015616	−.728264	.184468

−.204084	−.267982	−.511654
.468929	−.269317	.303203
−.415367	.415423	.040422
−.420326	−.185440	.392955
.465596	.435880	.007605
−.013102	.310084	−.048082
.069958	−.236415	.323698
−.056855	−.073670	−.275616
.067760	.428354	.030765
−.029680	−.210106	.319577
−.038080	−.218249	−.350342
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
−.396549	.161128	.285133

SCHEME 1. (Continued)

Imposing Constraints by Schmidt Orthogonalization
Number of Internal Coordinates Left in Active Space is 6
Schmidt-Orthogonalized Set of 6 Active and 3 Constraint Vectors

.000000	.000000	.000000	.000000	.000000	.000000
−.187075	−.208328	.668125	−.095975	−.386317	.562814
−.014591	.512617	.574657	.058420	.570263	−.095895
−.170009	−.411943	−.006245	−.787700	.311457	−.122119
−.128916	.597099	.019238	−.459672	−.564561	−.310573
−.738362	.007467	−.043225	.122218	.122905	−.086145
.182507	.186951	−.289352	−.033812	−.060619	.425984
.555855	−.194418	.332577	−.088406	−.062286	−.339839
−.126108	−.217709	.115690	.256622	−.208765	−.359376
.000000	.000000	.000000	.000000	.000000	.000000
.126108	.217709	−.115690	−.256622	.208765	.359376
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000

1.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	1.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	.000000	.000000
.000000	1.000000	.000000

Primitive Weights in Final Non-Redundant Optimization Space

Weight for primitive	1 is	.000000
Weight for primitive	2 is	1.000000
Weight for primitive	3 is	.931028
Weight for primitive	4 is	.931028
Weight for primitive	5 is	1.000000
Weight for primitive	6 is	.584566
Weight for primitive	7 is	.338264
Weight for primitive	8 is	.584566
Weight for primitive	9 is	.315274
Weight for primitive	10 is	.000000
Weight for primitive	11 is	.315274
Weight for primitive	12 is	.000000
Weight for primitive	13 is	.000000
Weight for primitive	14 is	.000000
Weight for primitive	15 is	.000000
Weight for primitive	16 is	.000000

SCHEME 1. (Continued)

Setting Up Default Hessian
Using Lagrange Multiplier Algorithm
Hessian Matrix

.224220	.007366	−.040438	.063083	.025134	−.012927
.007366	.449720	.050835	.029988	−.027778	−.090463
−.040438	.050835	.433109	−.010343	.017037	.094713
.063083	.029988	−.010343	.453318	.025371	.053801
.025134	−.027778	.017037	.025371	.467052	−.040442
−.012927	−.090463	.094713	.053801	−.040442	.331197
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000

.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	.000000
.500000	.000000	.000000	1.000000	.000000	.000000
.000000	.500000	.000000	.000000	1.000000	.000000
.000000	.000000	.200000	.000000	.000000	1.000000
1.000000	.000000	.000000	.000000	.000000	.000000
.000000	1.000000	.000000	.000000	.000000	.000000
.000000	.000000	1.000000	.000000	.000000	.000000

12 Hessian modes will be used to form the next step

Hessian Eigenvalues:

−.904988	−.780776	−.780776	.200000	.200000	.458617
.500000	.500000	.500000	1.104988	1.280776	1.280776

Hessian Eigenvectors:

.000000	.000000	.000000	.953512	.085772	−.140588
.000000	.000000	.000000	.012530	−.408205	.076672
.000000	.000000	.000000	.111511	.429749	.432893
.000000	.000000	.000000	−.251863	.222753	−.555507
.000000	.000000	.000000	−.055215	−.211130	.671502
.000000	.000000	.000000	.108351	−.739684	−.166036
.000000	.000000	−.615412	.000000	.000000	.000000
.000000	.615412	.000000	.000000	.000000	.000000
−.671005	.000000	.000000	.000000	.000000	.000000
.000000	.000000	.788205	.000000	.000000	.000000
.000000	−.788205	.000000	.000000	.000000	.000000
.741453	.000000	.000000	.000000	.000000	.000000

SCHEME 1. (Continued)

Hessian Eigenvectors:

−.105622	−.010425	.228974	.000000	.000000	.000000
−.826477	.310114	−.219314	.000000	.000000	.000000
−.246001	−.553823	−.498245	.000000	.000000	.000000
−.447104	−.470301	.396716	.000000	.000000	.000000
−.091398	−.211799	.669513	.000000	.000000	.000000
.192375	−.575290	−.213524	.000000	.000000	.000000
.000000	.000000	.000000	.000000	.000000	−.788205
.000000	.000000	.000000	.000000	.788205	.000000
.000000	.000000	.000000	−.741453	.000000	.000000
.000000	.000000	.000000	.000000	.000000	−.615412
.000000	.000000	.000000	.000000	.615412	.000000
.000000	.000000	.000000	−.671005	.000000	.000000

Minimum Search-Taking P-RFO Step
Searching for Lamda that Maximizes Along the Constraint modes Only
Value Taken Lamda = .06888687
Searching for Lamda that Minimizes Along All other modes
Value Taken Lamda = −.05842885
Calculated Step too Large. Step scaled by .796482
Step Taken. Stepsize is .300000

Parameter values and Displacements in Internal Coordinates

Coordinate	Current Value	Gradient	Displacement	New Value
1	−1.042211	.027646	−.063534	−1.105734
2	.760398	−.074441	.117816	.878213
3	2.891220	.003749	−.014338	2.876882
4	−2.508062	.090263	−.135811	−2.643873
5	−.429169	−.047291	.082038	−.347131
6	.481250	.065094	−.077066	.404183
7	2.645618	.085917	.139202	2.784820
8	4.535345	−.027587	.140844	4.676189
9	2.094395	−.026701	.039356	2.133751

Lagrange Multipliers for Constraints

Constraint	Current Value	Gradient	Displacement	New Value
1	.000000	−.188973	−.164903	−.164903
2	.000000	−.188973	−.058992	−.058992
3	.000000	−.052360	.015766	.015766

	Maximum	Tolerance	Cnvgd?
Gradient	.188973	.000300	NO
Displacement	.164903	.000300	NO
Energy change	−.024626	.000001	NO

SCHEME 1. (Continued)

four proper torsions, based on the standard connectivity (bonding) in this system. The nonbonded H—H distance is not a formal bond, and so this distance is added to the primitive set, giving a total of 16 primitive internals. Fluoroethylene has 12 degrees of freedom and so diagonalization of the 16×16 $\mathbf{B}\mathbf{B}^{\mathrm{T}}$ matrix gives four zero (corresponding to redundant eigenvectors) and 12 positive eigenvalues. Three of the 12 nonredundant eigenvectors can be eliminated due to symmetry (these are vectors with nonzero components for the torsions), leaving a set of nine active delocalized internal coordinates (the last component in each vector corresponds to the nonbonded H—H distance which was added to the initial primitive set).

The Schmidt-orthogonalization procedure isolates the three constraint coordinates, and our final Schmidt-orthogonalized set of delocalized internal

coordinates consists of six active and three (unit) constraint vectors. Note that the components in each of the active vectors corresponding to the primitives that are to be constrained are zero. Because none of the desired constraints are satisfied in the starting geometry, all three of the constraint vectors are active initially.

The initial (default) Hessian consists of the standard 9×9 Hessian that would be generated in an unconstrained optimization assuming all nine coordinates were fully active, padded out to 12×12 with rows and columns of unit vectors, with unit components corresponding to each constraint coordinate (i.e., Hessian components 7 through 9). Diagonalization of this Hessian matrix gives three negative eigenvalues (derived from the constraint Lagrange multipliers), and nine positive eigenvalues (from the nine active coordinates). The three lowest (negative) and the three highest modes contain only components derived from the constraint coordinate and its corresponding Lagrange multiplier. (This is not true on subsequent cycles, following the Hessian update; however, these modes are normally dominated by the constraints.)

The eigenvector following (EF) algorithm,[15] modified for use with Lagrange multipliers,[7] is used to determine the optimization step. On the first cycle (as noted above), the Lagrange multipliers are taken to be zero, and hence the gradient along each constraint is not modified from its ''real'' value. The Lagrange multiplier gradient is simply the actual value of the corresponding constraint; that is, the current coordinate value minus the desired constraint value; the ''new value'' for each Lagrange multiplier is the value it will take at the beginning of the *next* cycle, exactly like a ''real'' coordinate.

The just discussed constrained optimization of fluoroethylene converged in six cycles. After three cycles, all three constraints were satisfied and from cycle four onwards the three constraint coordinates were eliminated from the optimization space, and the standard EF algorithm[15] was used to calculate the step in the space of the six remaining active coordinates only.

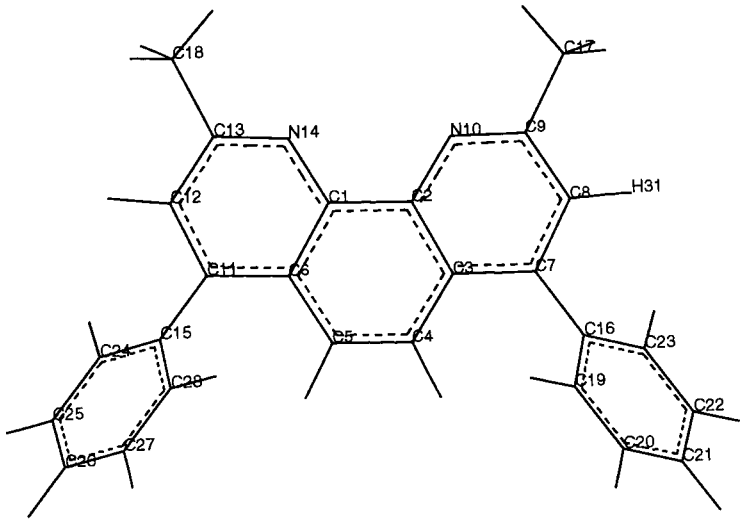
Examples

The Lagrange multiplier algorithm in delocalized internal coordinates described above has been implemented in the OPTIMIZE program.[16] This section presents optimizations, both constrained and unconstrained, on eight fairly large molecules, containing between 30 and 60 atoms. The molecules in the test set are: bathocuproin ($C_{26}H_{20}N_2$), BAO [2,5-bis-(4-aminophenyl)-1,3,4-oxadiazol] ($C_{14}H_{12}N_4O$), bispyrazolon ($C_{20}H_{18}N_4O_2$), cacothelin ($C_{21}H_{21}N_3O_7$), cis,cis-1,9-cyclohexadecadiene ($C_{16}H_{28}$), cadion ($C_{18}H_{14}N_6O_2$), 3,3′-dimethylnapthidine ($C_{22}H_{20}N_2$), and apigenin [4′,5,7-trihydroxyflavon] ($C_{15}H_{10}O_5$). A variety of constraints are imposed on these systems, comparing the performance of delocalized internal coordinates with Cartesians. Starting structures for each molecule, with appropriate atom labeling, are shown in Figure 1.

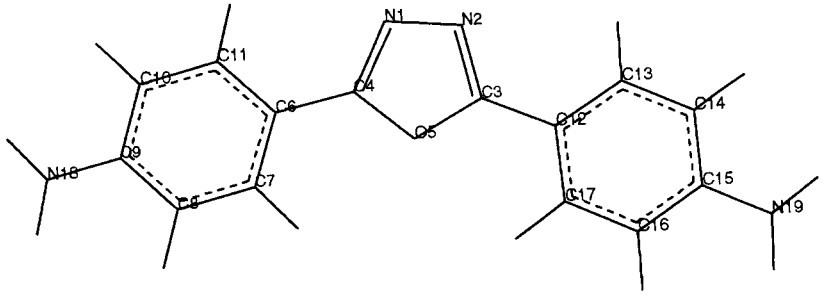
Bathocuproin (C_2) is a reagent for the determination of copper. BAO (C_{2v}) is used as a fluorescent stain for DNA. Bispyrazolon (C_2) is used for the determination of ammonia. 3,3′-Dimethylnapthidine (also C_2) is an indicator for complexometric titrations, especially involving zinc. Cadion (C_1) is a reagent for the determination of cadmium (II). Apigenin (C_s) is an inhibitor of human estrogen synthetase. Cis,cis-1,9-cyclohexadecadiene (C_i) and cacothelin (C_1) were chosen for their structural characteristics; the former is a single 16-membered ring and the latter contains a fairly complex, multiply fused ring system. All eight molecules were taken from the current Fluka catalogue.[17]

Energies and gradients for the above systems were calculated using the AM1[18] module in GAUSSIAN 94[19] and were then fed into OPTIMIZE for the optimization step. The molecules were constructed using the INSIGHTII graphical model builder,[20] and starting geometries in all cases were obtained by preoptimizing with the default mechanics forcefield in DISCOVER.[21]

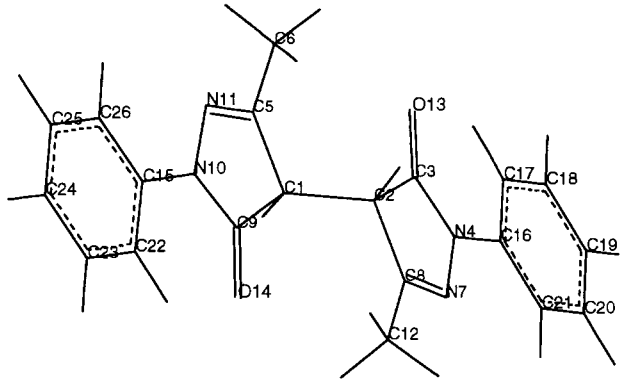
Optimizations were carried out using Cartesian and delocalized internal coordinates. Initial second derivative data was *not* provided; all optimizations started with a default Hessian (for details, see ref. 5), which was subsequently improved using the BFGS update.[22] Because previous work has shown that standard convergence criteria can cause Cartesian optimizations to converge prematurely (due to steps towards the end of the optimization that give a very small energy change), the convergence criteria were tightened to a maximum gradient component of 0.00005 a.u. (default 0.0003) and an energy change from the previous cycle of less than 10^{-7} hartrees (default 10^{-6}). This resulted in final energies (and structures) from Cartesian and internal coordinate optimizations that were generally in very good agreement with one another. The maximum allowed stepsize (in internal or Cartesian space where appropriate) was 0.3 a.u.



bathocuproin

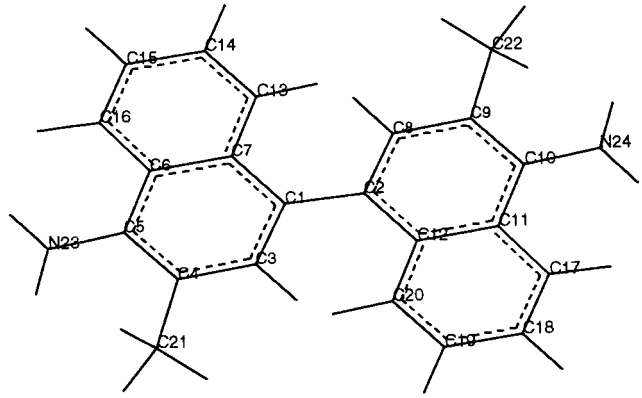


BAO

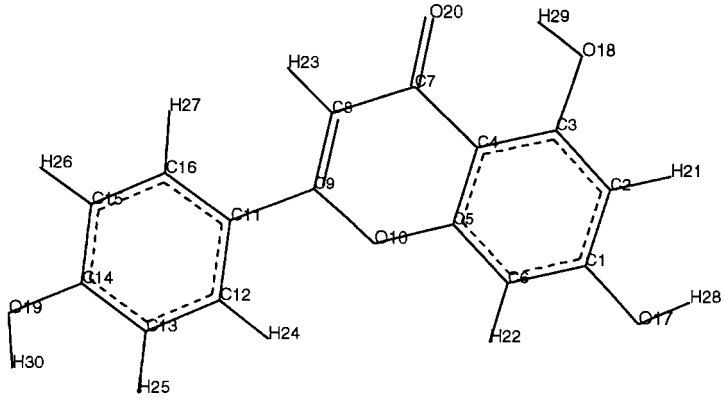


bispyrazolon

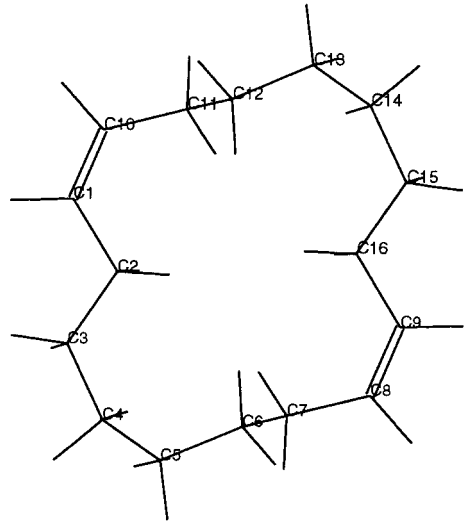
FIGURE 1. Schematic diagram of the eight molecules in the test set with appropriate atom labeling for the constraints listed in Table I.



3,3'-dimethylnapthidine

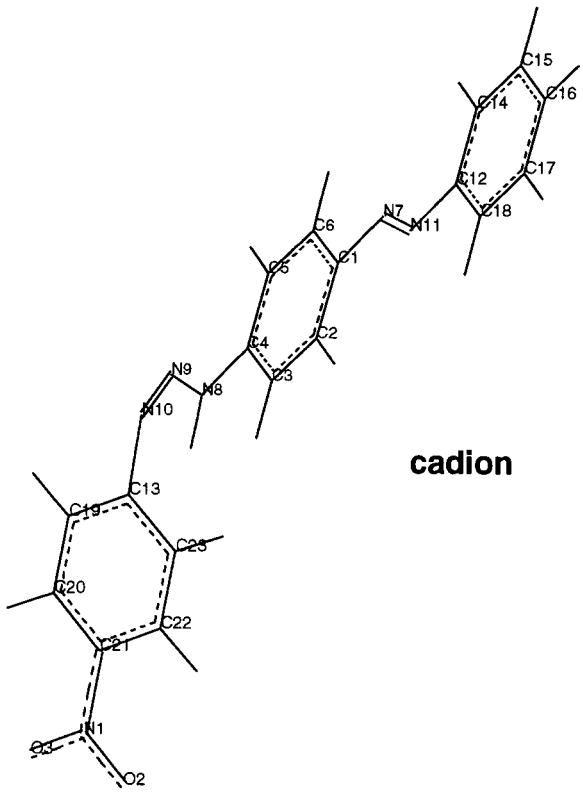


apigenin

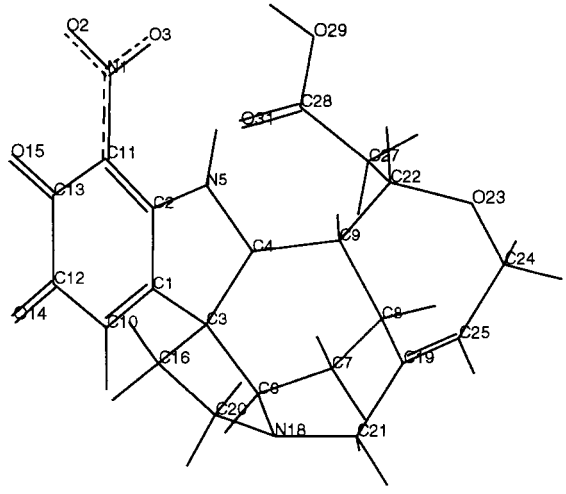


cis,cis-1,9-cyclohexadecadiene

FIGURE 1. (Continued)



cadion



cacothelin

FIGURE 1. (Continued)

TABLE I.

Number of Cycles to Reach Convergence for Constrained and Unconstrained Optimizations in Cartesian and Delocalized Internal Coordinates [Distance Constraints in Angstroms, Angle ($\angle$) and Torsion (τ) Constraints in Degrees].

System and constraints (starting values in parentheses)	Cycles to converge	
	Cartesian	Internal
Bathocuproin (C_2)		
Unconstrained	168	22
Constrained		
[$\tau C_{15}C_{11}C_7C_{16} = 20.0\ (15.0)$; $\tau C_{12}C_{11}C_6C_5 = 180.0\ (-174.1)$; $\tau C_{11}C_6C_5C_4 = 180.0\ (-179.3)$; $\tau C_3C_4C_5C_6 = 0.0\ (-1.9)$; $\tau C_5C_4C_3C_7$ and $\tau C_4C_3C_7C_8$ were also constrained by symmetry]	Failed[a]	18
BAO (C_{2v})		
Unconstrained	57	8
Constrained		
[$rC_7C_{17} = 5.0\ (5.418)$; $\angle C_7O_5C_{17} = 120.0\ (138.5)$]	49[b]	11
Bispyrazolon (C_2)		
Unconstrained	228	43[c]
Constrained		
[$rO_{13}O_{14} = 4.5\ (4.474)$; $\tau O_{13}C_2C_1O_{14} = 90.0\ (97.2)$]	127	43[c]
3,3′-Dimethylnapthidine (C_2)		
Unconstrained	151	42[c]
Constrained		
[$\tau C_3C_1C_2C_8 = 90.0\ (103.4)$]	147	44
Cadion (C_1)		
Unconstrained	323	32
Constrained		
[$rC_{12}C_{13} = 11.0\ (11.050)$; $\angle N_8N_7N_{11} = 125.0\ (131.9)$; $\tau C_{16}C_1C_4C_{21} = -45.0\ (-35.1)$]	233	26
Apigenin (C_s)		
Unconstrained	59	10
Constrained		
[$rO_{18}O_{20} = 2.7\ (2.739)$; $rH_{23}H_{27} = 2.2\ (2.185)$; $\angle C_{11}C_9C_{10} = 120.0\ (116.6)$]	43	9
Cis,cis-1,9-cyclohexadecadiene (C_i)		
Unconstrained	180	41
Constrained		
[$\angle C_4C_5C_6 = 120.0\ (131.9)$; $\tau C_2C_1C_{10}C_{11} = 0.0\ (-2.4)$; $\angle C_{12}C_{13}C_{14}$ and $\tau C_{16}C_9C_8C_7$ were also constrained by symmetry]	220[a]	37
Cacothelin (C_1)		
Unconstrained	208	58
Constrained		
[$rC_9C_{24} = 3.2\ (3.324)$; $\angle C_3C_8O_{23} = 120.0\ (124.0)$; $\tau C_1C_4C_7C_{21} = 120.0\ (124.5)$]	231	57

[a] Constraints in Cartesian coordinates imposed using dummy atoms (see ref. 13). Lagrange multiplier algorithm failed for bathocuproin; penalty functions not usable with dummy atoms.

[b] Lagrange multiplier algorithm failed; optimization converged using penalty functions. Constraints not well satisfied at convergence: $rC_7C_{17} = 4.998$; $\angle C_7O_5C_{17} = 121.3$.

[c] Internal coordinates converged to a lower energy than corresponding Cartesian optimization.

Results

Table I shows results in terms of the number of cycles to reach convergence. As can clearly be seen, optimization is far more efficient in delocalized internal coordinates than in Cartesians. This is true for both unconstrained and constrained optimizations. Optimizations in Cartesian coordinates for the systems examined here take between four and ten times *more* cycles to converge than the corresponding optimization in delocalized internals. This is fully in line with previous findings for unconstrained optimizations,[5] and it is pleasing to see the same factors in the constrained case as well. This provides very strong evidence of the efficiency of the Lagrange multiplier algorithm in delocalized internal coordinates. Note that the number of cycles required for convergence for the delocalized internal coordinate optimizations (both unconstrained and constrained) can be significantly reduced from the numbers quoted in Table I by using a better initial (default) Hessian guess[23]; this was not done in order to give a fairer comparison with the Cartesian optimizations, which used a simple unit matrix for the starting Hessian.

In addition to the improved efficiency, the internal coordinate algorithm is far more robust than its Cartesian equivalent. This can be seen in the two constrained optimizations in Cartesian coordinates in which the Lagrange multiplier algorithm failed (bathocuproin, which failed entirely, and BAO, which defaulted to the less accurate penalty function algorithm). The ability of the internal coordinate algorithm to cope smoothly with 0° and 180° dihedral angle constraints, without the necessity of introducing dummy atoms (bathocuproin and cis,cis-1,9-cyclohexadecadiene), is a further advantage. In addition, constrained optimizations in delocalized internal coordinates share with their unconstrained counterparts the tendency to converge to lower energy structures than the equivalent optimization in Cartesians,[3] as evidenced by the optimizations on bispyrazolon. (Note, however, that the *constrained* optimization on 3,3′-dimethylnapthidine converged to the *same* energy in both Cartesian and internal coordinates, even though the *unconstrained* optimization in Cartesians resulted in a higher energy structure than the internal coordinate optimization.)

Conclusions

By modifying a previous algorithm for constrained optimization using Lagrange multipliers in Cartesian coordinates,[7] a related algorithm that imposes constraints directly in delocalized internal coordinates has been presented. The new algorithm has all the advantages of delocalized internals in terms of efficiency—the rate of convergence compared to the Cartesian algorithm is four to ten times greater for the systems investigated in this work—and, in addition, is far more robust. Like the original Cartesian algorithm, distance, angle, and dihedral constraints can be imposed between *any* atoms in the molecule, irrespective of bonding, and imposed constraints *do not* have to be satisfied in the starting geometry. Unlike the Cartesian algorithm, 0° and 180° dihedral angle constraints can be imposed in a straightforward manner *without* introducing dummy atoms[13] or otherwise modifying the constraints.

Acknowledgments

The generous hospitality of Prof. Peter Pulay and the excellent working environment in his research group is gratefully acknowledged.

References

1. P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, *J. Am. Chem. Soc.*, **101**, 2550 (1979).
2. G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay, *J. Am. Chem. Soc.*, **114**, 8191 (1992).
3. J. Baker, *J. Comput. Chem.*, **14**, 1085 (1993).
4. J. Baker and W. J. Hehre, *J. Comput. Chem.*, **12**, 606 (1991).
5. J. Baker, A. Kessi, and B. Delley, *J. Chem. Phys.*, **105**, 192 (1996).
6. P. Pulay and G. Fogarasi, *J. Chem. Phys.*, **96**, 2856 (1992).
7. J. Baker, *J. Comput. Chem.*, **13**, 240 (1992).
8. R. Fletcher, *Practical Methods of Optimization: Constrained Optimization*, Vol. 2, Wiley, New York, 1981.
9. B. van de Graff and J. M. A. Baas, *J. Comput. Chem.*, **5**, 314 (1984).
10. E. B. Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations*, McGraw-Hill, New York, 1955.
11. C. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch, *J. Comput. Chem.*, **17**, 49 (1996).
12. J. Baker and P. Pulay, *J. Chem. Phys.* **105**, 11100 (1996).
13. J. Baker and D. Bergeron, *J. Comput. Chem.*, **14**, 1339 (1993).
14. S. Califano, *Vibrational States*, Wiley, London, 1976.

15. J. Baker, *J. Comput. Chem.*, **7**, 385 (1986).
16. J. Baker, OPTIMIZE, a suite of algorithms for geometry optimization.
17. Fluka Chemika-Biochemika Analytika Catalogue 1995/66, Fluka Chemical Co., Switzerland.
18. M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, and J. J. P. Stewart, *J. Am. Chem. Soc.*, **107**, 3902 (1985).
19. M. J. Frisch. G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. Keith, G. A. Petersson, J. A. Montgomery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. Defrees, J. Baker, J. J. P. Stewart, M. Head-Gordon, C. Gonzales, and J. A. Pople, *GAUSSIAN 94*, *Revision C.3*, Gaussian, Inc., Pittsburgh, PA (1995).
20. *INSIGHTII*, *Version 2.3.0*, Biosym Technologies, Inc., San Diego, CA (Dec. 1993).
21. *DISCOVER*, *Version 2.9.5*, Biosym Technologies, Inc., San Diego, CA (May 1994).
22. R. Fletcher, *Practical Methods of Optimization: Unconstrained Optimization*, Vol. 1, Wiley, New York, 1981.
23. H. B. Schlegel, *Theor. Chim. Acta*, **66**, 333 (1984).